CS

Hash Lab

1 Introducció

L'autenticació d'usuaris mitjançant contrasenyes és un dels mètodes més estesos a l'hora de protegir l'accés a sistemes informàtics. Els usuaris autoritzats a accedir al sistema disposen d'un compte (identificat per un nom d'usuari) i, per tal d'accedir al sistema, cal que introdueixin aquest identificador i una contrasenya. El sistema comprova si el parell usuari-contrasenya són vàlids i permet l'accés a l'usuari en cas afirmatiu.

L'autenticació per contrasenyes és simple (fàcil d'entendre per a usuaris no tècnics), ràpida, i no requereix d'infraestructura ni dispositius addicionals, però el seu ús no està exempt de desafiaments i riscos de seguretat.

Aquesta pràctica consta de dues parts. A la primera part, treballarem amb un conjunt de dades de contrasenyes reals que s'han fet públiques en diverses filtracions, i les utilitzarem per implementar un detector de contrasenyes filtrades. A la segona part, ens centrarem en atacs d'endevinació de contrasenyes fora de línia, en els quals l'atacant ha obtingut un fitxer de hashos de contrasenyes i l'objectiu de l'atac és endevinar quines contrasenyes conté. Farem servir les eines més populars actualment per tal de trencar els hashos de diverses contrasenyes.

2 Part I: Filtres de Bloom

Els filtres de Bloom són estructures de dades probabilístiques que permeten comprovar de manera eficient si un element és membre d'un conjunt. Són especialment útils en situacions on l'espai de memòria és limitat, ja que ofereixen una representació compacta del conjunt. Tot i que poden produir falsos positius, mai generen falsos negatius, cosa que els fa adequats per a aplicacions on és preferible un petit marge d'error a canvi d'una major eficiència.

En el context de la seguretat de contrasenyes, els filtres de Bloom poden ser utilitzats per comprovar ràpidament si una contrasenya ha estat compromesa sense necessitat de mantenir una llista completa de totes les contrasenyes filtrades. Això permet una verificació ràpida i eficient, fins i tot amb conjunts de dades molt grans, com el que utilitzarem en aquest laboratori.

El conjunt de dades que farem servir per a aquesta part de la pràctica és una de les bases de dades de contrasenyes en clar més grans conegudes actualment[1]. Ocupa 41GB i conté 1400 milions de credencials, agregades a partir de dades de diverses filtracions. El conjunt de dades conté contrasenyes de multitud de serveis diferents, com ara LinkedIn, Netflix o PayPal.

Cadascuna de les entrades del conjunt de dades és un parell format per un correu electrònic i una contrasenya. Per exemple:

[1] https://medium.com/4iqdelvedeep/1-4-billion-clear-text-credentials-discovered-in-a-single-database-3131d0a1ae14

```
SAP-Recruitment@live.com.sg:level25
SAP-ZTP-300OOFOS08@mail.ru:0815866
SAP-ben@hotmail.com:19851014
SAP-not_alone@live.fr:audrey87
SAP.07@MAIL.RU:000086
SAP.72@MAIL.RU:LALA
SAP.BPC.75.NW@gmail.com:9959467196
SAP.CDC@hotmail.com:pakistan@1
SAP.MIKE@yahoo.com:tigerbird
```

Com que el conjunt de dades és gran, les dades es troben emmagatzemades en diversos fitxers, organitzats per carpetes en funció de les primeres lletres del correu electrònic de l'usuari.

Descarregueu el dataset i familiaritzeu-vos amb la seva estructura.

Exercici 1.- Implementació filtre de Bloom

Implementeu un filtre de Bloom fent servir funcions hash independents. Com a mínim, és necessari poder especificar la mida del filtre i el número de funcions hash, i cal poder afegir nous elements al filtre i comprovar la inclusió d'elements al filtre. Addicionalment, implementeu funcions per desar el filtre a disc i carregar un filtre desat.

Exercici 2.- Impacte del número de funcions hash en la probabilitat de falsos positius

Seleccioneu un subconjunt de contrasenyes del dataset total i fixeu una mida per al filtre de Bloom. Creeu diversos filtres de Bloom amb diferents quantitats de funcions hash (diferents valors de k) i afegiu els elements del subconjunt de contrasenyes seleccionat a cadascun dels filtres.

Calculeu la taxa de falsos positius experimental i teòricament per a cada un dels filtres i genereu una gràfica que mostri com evoluciona la probabilitat de falsos positius (a nivell teòric i experimental) en funció de k.

Exercici 3.- Selecció òptima de paràmetres del filtre

Implementeu una funció que seleccioni la mida òptima del filtre i el nombre de funcions de hash a partir del nombre esperat d'elements i la taxa de falsos positius desitjada.

Exercici 4.- Comparativa: filtre de Bloom vs conjunt

Compareu experimentalment la implementació del filtre de Bloom amb una estructura de dades no probabilística estàndard (per exemple, un conjunt o una llista).

Compareu les dues estructures en base a les 4 mètriques següents:

- temps d'afegir elements al filtre
- temps de comprovar la pertinença d'elements al filtre
- mida de l'estructura de dades
- taxa de falsos positius

Feu la comparació per a 3 conjunts de dades de contrasenyes de diferents mides (per exemple, 1K, 10K, 1M contrasenyes). Podeu agafar subconjunts de contrasenyes del dataset total.

Comenteu els resultats obtinguts.

Exercici 5.- Optimització del filtre amb doble hashing

Com haureu pogut observar a l'exercici anterior, el filtre de Bloom és lent afegint elements i comprovant-ne la pertinença al filtre. Una part significativa del temps dedicat a afegir i comprovar elements s'inverteix en el càlcul de les diverses funcions hash.

Implementeu una nova versió del filtre de Bloom que funcioni amb doble hashing. El doble hashing utilitza dues funcions de hash, h_1 i h_2, per generar les múltiples funcions hash necessàries, g_i, per a implementar el filtre, de manera que:

$$g_i(x) = h_1(x) + i \cdot h_2(x) \bmod m$$

on h_1 i h_2 són funcions de hash independents, i és l'índex de la funció de hash, i m és la mida del filtre.

Exercici 6.- Comparativa: funcions hash independents vs doble hashing

Compareu la nova implementació del filtre de Bloom amb la primera implementació (és a dir, la implementació que fa servir la tècnica del doble hashing amb la implementació original, que fa servir funcions hash independents). Utilitzeu les mateixes 4 mètriques i els mateixos 3 subconjunts de dades utilitzats a l'exercici 4.

Comenteu els resultats obtinguts.

Exercici 7.- Filtre de Bloom per al dataset complet

Donats els resultats de la comparació anterior, creeu un filtre de Bloom per a tot el conjunt de dades de contrasenyes amb la millor configuració possible, suposant que podeu tolerar com a màxim una taxa de falsos positius del 5%. Responeu les següents preguntes:

1. Quina versió de la implementació heu utilitzat?
2. Quins són els paràmetres (mida i número de funcions hash) del filtre?
3. Quants elements heu afegit al filtre?
4. Compareu l'espai necessari per emmagatzemar el filtre amb la mida del conjunt de dades.
5. Quant temps ha estat necessari per afegir totes les contrasenyes al filtre?
6. Indiqueu quines de les contrasenyes següents s'han filtrat (és a dir, es troben al filtre).

```
hola
1234
iloveyou
Awesome1
mmmmmmm
367026606991464
supertrooper2002
SpRyhdjd2002
593b04318425a33190ceaabab648376c
bnbd246GbB
```

3 Part II: Hash *cracking* de contrasenyes

Emmagatzemar les contrasenyes dels usuaris en clar és una pràctica de seguretat extremadament perillosa, ja que les fa vulnerables a diversos tipus d'atacs. Per exemple, un administrador del sistema que tingui accés a les contrasenyes les pot veure (cosa que ja de per sí pot comportar problemes de privadesa) i fer-les servir per suplantar els usuaris en altres serveis. A més, si el sistema és compromés, l'atacant recuperaria també totes les contrasenyes.

Per això, es fan servir diverses tècniques per emmagatzemar contrasenyes, la més popular de les quals és l'ús de funcions hash. Així, per cada usuari autoritzat, s'emmagatzema el seu nom d'usuari i el hash de la seva contrasenya. Això permet verificar que la contrasenya introduïda per l'usuari durant el procés d'autenticació és correcta sense desar directament aquesta contrasenya.

En aquesta part de la pràctica, ens centrarem en **atacs d'endevinació de contrasenyes fora de línia**. En aquest tipus d'atacs, assumim que l'atacant ha obtingut un fitxer de hashos de contrasenyes (per exemple, el fitxer `/etc/passwd` per a contrasenyes d'accés a sistemes unix) i l'objectiu de l'atac és endevinar quines contrasenyes conté.

Per fer-ho, farem servir el coneixement que hem adquirit sobre les contrasenyes més comunes i els patrons habituals en la construcció de contrasenyes que hem vist a l'apartat anterior, i veurem diverses eines que permeten executar els atacs.

El primer pas per atacar un hash és identificar quina funció l'ha produït, procés que es realitza en base al seu format i la seva longitud. Tot i que el procés es pot fer manualment, existeixen diverses eines que el faciliten, informant de les possibles funcions hash que poden haver generat un hash donat, així com dels identificadors d'aquestes funcions en diversos programes que permeten atacar-los. Per fer aquesta activitat, us recomanem l'ús de l'eina haiti (a la documentació s'hi detalla el procés d'instal·lació i la sintaxi bàsica per utilitzar-la).

Una vegada deduïda la funció hash, el següent pas consisteix a intentar trobar la contrasenya. Com que les funcions hash són unidireccionals, aquest pas necessàriament requereix d'un procés de prova i error, informat a partir del coneixement de l'estructura de la contrasenya, les contrasenyes més comunes, informació contextual sobre l'usuari, etc. John the Ripper és un programa *open source* que es considera un estàndard de facto per trencar hashos. John és capaç de realitzar atacs de força bruta, atacs de diccionari, i atacs basats en patrons (que poden arribar a ser molt complexos). Això la fa una eina molt versàtil.

Exercici 8.- Atacs de força bruta

Recupereu les preimatges dels hashos següents fent servir un atac de força bruta, sabent que aquestes estan formades només per dígits. Per cada hash, proporcioneu la funció hash utilitzada i la preimatge. Addicionalment, indiqueu la comanda que heu fet servir per obtenir les preimatges.

```
h4_1:  6ea9ab1baa0efb9e19094440c317e21b
h4_2:  8e296a067a37563370ded05f5a3bf3ec
h4_3:  54fe976ba170c19ebae453679b362263
h4_4:  6562c5c1f33db6e05a082a88cddab5ea
h4_5:  7c590f287acefdd3ea84a7678f1e907b
h4_6:  6593a1651adf82783394195112e73aac
h4_7:  a388742c988cb1b8d9a304db528cf71d
h4_8:  047d8415eec2dcec989c77d531535531
```

```
h4_9:  b87262873e28e7589c15c5e467e9c39a
h4_10: 42005f9a3f3a28aabe4883bb7a60ec0a
h4_11: 1f99c8a687de5b829addfce79383827a
h4_12: 1d1803570245aa620446518b2154f324
```

Exercici 9.- Atacs de força bruta - mida de l'alfabet

Calculeu quin és el número màxim de hashos que cal fer per assegurar que recuperem una contrasenya per cadascuna de les longituds de les contrasenyes de l'exercici anterior, si aquesta consta únicament de caràcters numèrics. I si la contrasenya és alfanumèrica?

Calculeu quants hashos sha-1 per segon és capaç de calcular la màquina en què estigueu fent aquesta pràctica. Quant temps tardaríeu, com a màxim, en trobar una contrasenya de la longitud indicada per als dos alfabets?

A partir de les respostes als dos subapartats anteriors, raoneu sobre la importància de la longitud d'una contrasenya i la mida de l'alfabet que s'utilitza.

Exercici 10.- Atacs de diccionari

Els atacs de força bruta poden ser útils quan l'espai de possibles contrasenyes és reduït, però són computacionalment massa costosos quan aquest augmenta. Una alternativa a aquests són els atacs basats en diccionari.

Realitzeu atacs basats en diccionari per recuperar les preimatges del hashos següents. Per cada hash, proporcioneu la funció hash utilitzada, la preimatge i el diccionari utilitzat. Les contrasenyes d'aquest exercici són contrasenyes comunes, així que les trobareu en diccionaris genèrics. Addicionalment, indiqueu la comanda que heu fet servir per obtenir les preimatges.

```
h6_1: baf48e6b76ed3f590dba965a70603098
h6_2: ae57ac4302b0fbbc9ff3941f9c6809b3d36283226aba0ba5c6bd1b393df537ec
h6_3: d7ff97b5fcc6efb32095bec245e0437ba12965346983f7cf64cb1ed9d4ee5db4d5c9855
      caf850e4a70b49425408ab6a5a652f39a2ecccc6c8162c39f83b41b76
h6_4: $1$1234$BPwMGGSXa77b3Tul1zGYV0
h6_5: $5$1234$AEx6b7jdHN0aR3Aw96VUxKmCM9t1nl.r/OnvoOaThP8
```

Quina de les preimatges ha costat més temps de trobar? Justifiqueu-ne els motius.

Exercici 11.- L'ús de sal

Realitzeu un atac de diccionari contra els dos fitxers de contrasenyes que trobareu a continuació. Per cada hash, proporcioneu la funció hash utilitzada, la preimatge i el diccionari utilitzat.

Indiqueu el temps que ha estat necessari per fer els dos atacs i justifiqueu el resultat.

```
$1$apinchof$SRLUzGoq4VrvhxONRvMxG1
$1$apinchof$I/jqECZB5870b09axDIOOO
```

```
$1$1234$Gn/iq6aMn4fd2jJ55dNex.
$1$4321$ichFZkZw9YGuvao32tcvl1
```

(Opcional) Exercici 12.- Atacs de diccionari amb informació contextual

Sovint els usuaris escullen contrasenyes que no són comunes en relació a les contrasenyes triades per altres usuaris, però que són fàcilment deduïbles per un atacant que disposa d'informació contextual. En aquest exercici, executarem atacs de diccionari fent servir diccionaris construïts manualment a partir de la informació contextual que es disposa sobre el propietari de la contrasenya que es vol atacar.

Per cada hash, proporcioneu la funció hash utilitzada, la preimatge i el diccionari utilitzat. Expliqueu com heu construït el diccionari.

Nota: Podeu fer servir l'estratègia que considereu oportuna per aconseguir els diccionaris. Algunes alternatives són construir-los manualment, fer servir diccionaris de tercers (que trobareu a Internet) o bé fer servir eines que us ajudin a trobar-los o construir-ls. En aquest sentit, us recomanem que feu un cop d'ull les eines wordlistctl, CeWL i pnwgen.

```
h8_1: ae2bfa7cec2b567b01f62f8cd59d579bd445587f3fc66c56b5341d081f4e1901
h8_2: 5f541a0b1b477d891d7a26435125f999cb563c33e2b65f2a86aa9931f6520553
h8_3: dc75c9921b47d40ee704c58778a2cc1f6f4f5f4c144f019ba897ffe3613c820d
```

La informació contextual que disposeu per a fer l'exercici és la següent:

1. El hash $h8_1$ correspon a la contrasenya de la Marta, una noia que està enamorada de la **ciutat** on viu.
2. El hash $h8_2$ correspon a la contrasenya d'en John Smith, una noi de parla anglesa molt aficionat als **esports**. Sempre fa servir com a contrasenya el nom del següent esport que té pendent de provar.
3. El hash $h8_3$ correspon a la contrasenya d'en Marc, un mestre català que acaba d'aprovar les oposicions i està pendent que li assisnin **escola de destí**.

(Opcional) Exercici 13.- Atacs de diccionari amb alteracions i informació contextual

Com hem vist a la primera part de la pràctica, molts usuaris alteren les seves contrasenyes per intentar dificultar atacs com els que estem fent en aquesta pràctica. Ara bé, també hem vist que aquestes alteracions són bastant predictibles... les persones tendim a fer els mateixos canvis a les contrasenyes!

En aquest exercici, executarem atacs de diccionari fent servir informació contextual i alteracions comunes a les contrasenyes.

Per cada hash, proporcioneu la funció hash utilitzada, la preimatge, el diccionari base utilitzat i la comanda o estratègia seguida per introduir alteracions a la contrasenya.

```
h9_1: adacf9fd76805303d9126dbbfeb9262809b8306e5010027b6337a8278efa0e67
h9_2: 38c604df90a01b3c900ea55791b98265bf4669d101ffc225c681a936e75f1945
h9_3: e3a711431dfe615927f965e794f09380a17f8da484568e15ee236c1a94dfc124
```

La informació contextual que disposeu per a fer l'exercici és la següent:

1. El hash $h9_1$ correspon a una de les 100 contrasenyes més utilitzades pels usuaris d'Adobe, concatenada amb un any i un símbol.
2. El hash $h9_2$ correspon a la contrasenya d'en John Smith (veure exercici anterior), que ha alterat una mica el seu patró de creació de contrasenyes, i ara substitueix alguns caràcters

per símbols o números que s'hi assemblin. Per exemple, en comptes d'escriure *football*, escriuria *f00tb4ll* o potser *f()()tb@ll*.

3. El hash $h9_3$ correspon a la contrasenya d'en Pere, que sempre fa servir com a contrasenya patrons donats per la disposició de les tecles d'un teclat. Aquesta vegada, però, es va equivocar en una lletra a l'hora d'introduir la contrasenya!

4 Lliurament

Cal que lliureu un únic fitxer comprimit que contingui:

- El codi de la vostra implementació.
- Un PDF amb la memòria amb la informació que es detalla a continuació.

1. Dades personals: nom i cognoms dels integrants del grup. Només es lliurarà un informe per grup.
2. Les respostes a les preguntes plantejades en aquest document:
   (a) Per als exercicis 2, 4 i 6: les gràfiques comentades. Cal incloure les gràfiques resultants i cal explicar què s'observa a les gràfiques i perquè els resultats obtinguts són coherents amb el que s'espera a nivell teòric.
   (b) Per a l'exercici 7: repostes a les preguntes plantejades.
   (c) Per als exercicis de la part II (8-13), les respostes a les preguntes plantejades i el procediment que heu fet servir per aconseguir-les (incloent, si n'és el cas, el codi o les crides als scripts).